M. Conclusion

[0277] The distributing computing services platform, comprising programming model, schema, user interface, events, messaging, storage, directory, security, and code management component, facilitates more Internet based collaboration, and more inter-Web site communication.

[0278] The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

What is claimed and desired to be secured by United States Letters Patent is:

1. In a server federation that includes a plurality of servers that communicate with a plurality of client devices, a method for fulfilling a request comprising the following:

a first server of the plurality of servers receiving a request for a service;

the first server determining that further information is needed from at least a second server in order to respond to the request for the service; the first server structuring a request for the further information utilizing a schema recognized by both the first server and the second server, wherein the meaning of the request for further information is implied by the schema;

the first server dispatching the request for the further information to the second server using a transport-independent messaging infrastructure; the first server receiving a response from the second server, the response including the further information; and

the first server using the further information to respond to the request for the service.

2. A method in accordance with claim 1, wherein the first server structuring a request for further information comprises the following: the first server structuring the request in accordance with an eXtensible Markup Language (XML).

3. A method in accordance with claim 1, wherein the first server receiving a response from the second server comprises the following:

the first server receiving the response from the second server in the form of a data structure structured in accordance with an extensible Markup Language (XML).

4. A method in accordance with claim 1, wherein the first server dispatching the request for the further information to the second server using a transport-independent messaging infrastructure comprises the following:

the first server dispatching the request for the further information using a Simple Object Access Protocol (SOAP).

5. A method in accordance with claim 4, wherein the first server structuring a request for the further information comprises the following:

the first server structuring the request in accordance with an extensible Markup Language (XML).

6. A method in accordance with claim 1, wherein the first server receiving a response from the second server comprises the following:

the first server receiving the response from the second server using a Simple Object Access Protocol (SOAP).

7. A method in accordance with claim 6, wherein the first server receiving a response from the second server further comprises the following:

the first server receiving the response from the second server in the form of a data structure structured in accordance with an extensible Markup Language (XML).

8. A method in accordance with claim 1, wherein the first server dispatching the request for the further information to the second server using a transport-independent messaging infrastructure comprises the following:

a messaging component communicating with a common Application Program Interface that supports a plurality of transports.

9. A method in accordance with claim 8, wherein the first server dispatching the request for the further information to the second server using a transport-independent messaging infrastructure comprises the following:

the common Application Program Interface communicating with an appropriate one of the plurality of transports in response to the messaging component communicating with the common Application Program Interface.

10. A method in accordance with claim 9, wherein the common Application Program Interface communicating with an appropriate one of the plurality of transports comprises the following:

the common Application Program Interface communicating with a HyperText Transport Protocol (HTTP) transport.

11. A method in accordance with claim 9, wherein the common Application Program Interface communicating with an appropriate one of the plurality of transports comprises the following:

the common Application Program Interface communicating with an MSMQ binary transport.

12. A method in accordance with claim 8, wherein the common Application Program Interface communicating with an appropriate one of the plurality of transports comprises the following:

the common Application Program Interface communicating with a multicast transport.

13. A method in accordance with claim 8, wherein the common Application Program Interface communicating with an appropriate one of the plurality of transports comprises the following:

the common Application Program Interface communicating with an SMTP transport.

14. A method in accordance with claim 1, wherein the first server of the plurality of servers receiving a request for service comprises the following:

the first server of the plurality of servers receiving the request for service from a third server.